POLITECNICO DI TORINO

Corso di Laurea Magistrale in Ingegneria Informatica

Master Thesis

Pin Control Protection

Protecting Linux Kernel Pin Control Subsystem from Pin Control Attack

Supervisor
prof. Antonio Lioy

Candidate
Andrea Genuise

Academic Year 2016-2017

† A nonno Carmelo

Summary

Recently embedded systems have become more and more integrated with all aspects of our lives, and their security concerns have risen as well. They spread in various fields such as electronic devices, home automation, manufacturing and mission critical applications. These systems, in particular PLCs deployed within the context of an Industrial Control System, use Input/Output interfaces to interact with the physical world by means of sensors and actuators. As demonstrated by a novel kind of attack called Pin Control Attack, one can tamper with the integrity or the availability of legitimate I/O operations, factually changing how a PLC interacts with the outside world and possibly causing physical damage to people and environment. In this thesis we design and implement a possible countermeasure to the attack for Linux Kernel on an ARM-based Programmable Logic Controller, showing its effectiveness and impact on embedded systems which usually have very limited resources.

Acknowledgements

TODO Acknowledgements.

Contents

Chapter 1

Introduction

Embedded systems are widely used today in various applications, from cars, cell phones, home automation, to critical infrastructures like power plants and power grids, water, gas or electricity distribution systems and production systems for food and other products. Within the context of an Industrial Control System (ICS), they are better known as PLCs (Programmable Logic Controllers).

Since they were almost isolated from the network, embedded systems have not been built with security in mind. However, many recent cyber-attacks demonstrated that such an assumption is no more valid, and the security of embedded systems became an open question to deal with.

Unfortunately, this could be a more difficult long-term problem with respect to security for desktop and enterprise computing, both for the limited capabilities of these systems and for the physical side effects a security breach may lead to, including property damage, personal injury, death and even environmental or nuclear disaster.

The PLCs control the outside world via their I/O interfaces: therefore, they must be both reliable and secure. Digging into their architecture, we know that the I/O interfaces of PLCs (e.g. GPIO, SCI, JTAG, etc.), are usually controlled by a so called System on a Chip (SoC), an integrated circuit that combines multiple I/O interfaces. In turn, the pins in a SoC are managed by a pin controller, a subsystem of SoC, through which one can configure the operating mode of the pins, such as the input or output mode. One of the most peculiar aspects of a pin controller is that its behavior is determined by a set of registers: by altering these registers one can change the behavior of the chip in a dramatic way. In [1], Abbasi et al. showed how this feature is exploitable by attackers, who can tamper with the integrity or the availability of legitimate I/O operations, factually changing how a PLC interacts with the outside world.

Based on these observations, they introduced a novel attack technique against PLCs, called Pin Control Attack. The salient features of this new class of attacks are:

1. It is intrinsically stealth. The alteration of the pin configuration does not generate any interrupt, preventing the Operating System (OS) to react to it.

2. It is entirely different in execution from traditional techniques such as manipulation of kernel structures or system call hooking, which are typically monitored by anti-rootkit protection systems.

3. It is viable. It is possible to build concrete attack using it.

To better understand a possible attack scenario, Fig. 1.1 shows a simplified control system. The system consists of a boiler equipped with a relief valve driven by the PLC according to the value provided by a temperature sensor. The PLC is connected to a SCADA server (Supervisory Control And Data Acquisition), which usually comes with a control software used to program the PLC itself and to monitor its current status from a terminal. Suppose that the PLC normally

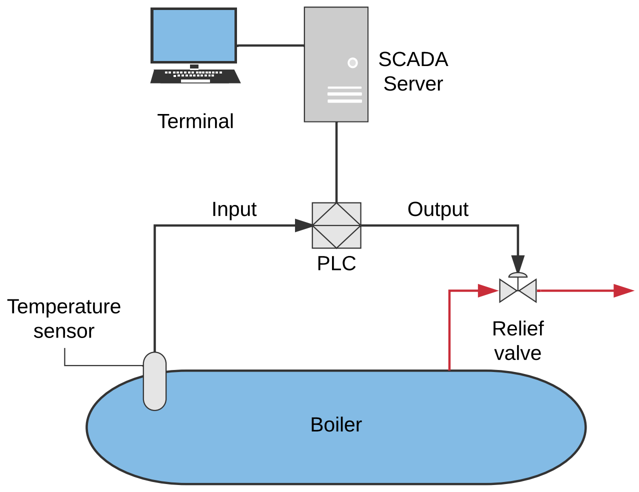

Figure 1.1. Simplified control system: a possible target of Pin Control Attack.

reads the values from the temperature sensor, and is programmed to open the relief valve if the temperature goes over 80 C. An attacker could simply tamper the temperature value read by the PLC, reconfiguring the pin related to the sensor as output and writing its own desired value (e.g. always 50 C) regardless of the actual temperature. Actually there is no detection mechanism able to react to this configuration change, neither in the PLC firmware nor in the control system. Moreover, the operator of the control system will not be able to see the real temperature from the terminal, but only the one forged by the attacker, so it may likely not notice the intrusion at all. Such a condition may lead to an uncontrolled overheating, and if it is not detected in time it could even make the boiler to explode.

To overcome this sophisticated attack, we propose a defensive mechanism to extend the existing Linux Kernel Pin Control Subsystem, frequently used in embedded systems, to be able to detect our novel attack. It is a challenging task for two reasons:

1. The Pin Control lacks hardware interrupts, so it is not possible to directly react to any configuration change. More complex detection mechanisms are needed in order to achieve the highest possible detection rate.

2. The resources available within an embedded system like a PLC are extremely limited. Therefore, our solution must be extremely agile and light since the smallest delay in the PLC I/O operation may have unintended consequences for the controlled process.

In Chapter 2 we list the known attacks to embedded systems existing in the literature, and then discuss the protection mechanisms currently available. As Pin Control Attack is a new kind of attack, to the best of our knowledge, at the time of writing, no one of the existing protections is able to prevent nor detect it. In Chapter 3 we present the architecture of our proposed Pin Control Protection, then we describe the implementation and provide developer and user manual for it. The Chapter 4 provides the results obtained during the experiments, showing the detection rate and the performance overhead on a PLC environment. Finally, in Chapter 5 we analyse the shortcomings of our defense and the possible future works and improvements.

Chapter 2

Related work

In this chapter we summarise the state of the art about security of embedded systems at the time of writing. First, we discuss about the attacks of the recent years, showing how the embedded systems security concerns are rising. Next, we analyse the defense mechanisms currently available in the literature, realising that they are still in a very early stage of their life.

2.1 Attacks

In the past few years a lot of attacks targeted embedded systems: most notably the infamous Stuxnet [2] targeting an Iranian nuclear facility in 2010. More recently Grandgenett et al. [3] analysed the authentication protocol between the RSLogix 5000 software and the PLC, based on a simple challenge-response mechanism. Since the protocol lacks freshness in its messages, is vulnerable to replay attacks, through which an attacker could repeat privileged commands to the PLC. Furthermore, they found that both the decoding of the challenge and the encoding of the response use an RSA-2048 key which is hard-coded in the RSLogix software, and it is actually valid for any Rockwell/Allen Bradley PLC. This indicates how the security mechanisms of these systems often have a really poor design, if any.

Papp et al. [4] analysed the existing attacks on embedded systems, relying on the proceedings of security conferences, with a focus on computer hacking, and on the Internet for media reports, blogs and mailing list. They built a taxonomy based on five dimensions: precondition, vulnerability, target, attack method and effect of the attack.

For our purpose, we may classify the attacks found in literature using a simpler criterion based on the attack method. We may distinguish three main categories:

1. **Firmware modification**: all the attacks aiming to upload a malicious firmware version (or part of it) in the device belong to this category.

2. **Logic modification**: this category consists of the attacks that modify the PLC logic in some way. In this case the integrity of the firmware is not violated, but a malicious program, or logic, is inserted into the PLC to make it misbehave during the control process.

3. **Control flow modification**: it includes the attacks that alter the normal control flow of a process by leveraging classic programming vulnerabilities such as buffer overflow or expired pointer dereference.

We briefly report about these different kind of attacks in the following sections.

2.1.1 Firmware modification attacks

Almost all modern embedded systems provide a way to update the firmware, and the attackers could exploit this feature to upload its own malicious firmware. Basnight et al [5] reverse engineered an Allen-Bradley ControlLogix L61 PLC firmware showing how to bypass the firmware update validation method and successfully upload a counterfeit firmware. Peck et al. [6] demonstrate how using commonly available software an attacker can write and load his malicious firmware into Ethernet cards of devices used in control systems, potentially infecting the entire industrial control system. Cui et al. [7] discovered a vulnerability in the HP-RFU (Remote Firmware Update) feature of LaserJet printers, that allows remote attackers to make persistent modifications to the printer's firmware by simply printing to it.

2.1.2 Logic modification attacks

Stuxnet [2] belongs to this category. Along with its several components, mainly used to replicate and control the malware, its core is essentially an infected version of a SCADA software library used to program the PLC itself. By hooking some of the library functions it is able to load infected code and data blocks into the PLC and hide itself from the operator. McLaughlin et al. [8, 9] evaluated some techniques and implemented a tool (SABOT) to infer the structure of a physical plant and craft a dynamic payload, allowing an attacker to cause an unsafe behavior without having a deep *a priori* knowledge of the target physical process. Similar techniques might mitigate the precondition needed by an attack, making it even more viable. Beresford [10] showed how the PLCs and the protocols used for communication in control systems were built without any security in mind, and demonstrated that they are affected by many vulnerabilities which may also enable the attacker to know the current configuration and rewrite the PLC logic. More recently, Klick et al. [11] used an internet-facing PLC as a network gateway by prepending a backdoor, made of a port scanner and a SOCKS proxy, to the existing logic code of the PLC.

2.1.3 Control flow modification

Many recent advisories [12–15] from ICS-CERT (Industrial Control System Cyber Emergency Response Team) report about various programming vulnerabilities affecting both PLC firmwares and control softwares. Most of them allow remote code execution and could be exploited without requiring particularly high skills. The vulnerabilities discovered by Beresford [10] also allow the attacker to insert a payload into the logic and subvert the control flow to execute malicious code. Nevertheless, the majority of the PLCs run the applications with root privileges, so it is quite simple for an attacker to get a root shell. One of the most dangerous kind of control flow attacks consists of ROP (Return-Oriented Programming) techniques, or similar variants [16, 17] which leverage different sequence of instructions equivalent to a return instruction. Since code vulnerabilities may affect embedded systems [10, 12–15], ROP techniques are applicable as well. Furthermore, due to the limitations imposed by these systems, is even more challenging to defend against them.

2.2 Defenses

Chapter 3

Pin Control Protection Design

NB: Just draft chapter.

In this chapter we first analyse Pin Control Attack in more detail, and then show the design and the implementation of the proposed defense.

3.1 Pin Control Attack Analysis

First, we try to better classify it in order to better understand which type of attack we are going to address.

We assert that our attack cannot be completely categorised using the taxonomy proposed in [4], resulting in at least two dimensions with *Unknown* value.

1. **Precondition**: we have at least two different variants of Pin Control Attack, which will be discussed in more detail in Chapter 3, and they need different precondition. Both need at least *local or remote access* to the system, and an a priori knowledge of the physical process.
2. **Vulnerability**: in our case the exploited vulnerability is basically the lack of hardware interrupts as reaction to hardware configuration changes. This is about the communication between hardware and operating system and, to the best of our knowledge, it has never been addressed before.
3. **Target**: the target layer resides between hardware and firmware/OS. They suggest the generic *device* target for this case.
4. **Attack method**: our attack installs a *malware* inside the PLC. More specifically it is a rootkit that could not be detected with the current available protections, if any.
5. **Effect**: our most dangerous effects are not about code execution, integrity violation or denial of service as for the most of the existing attacks. Our main effects are directly propagated into the physical world surrounding the violated system, as you may notice from example in Chapter 1, and they strongly depends on which type of process the PLC is actually controlling.

The classification of Pin Control Attack based on the taxonomy proposed by Papp et al. [4] is summarised in Tab. 3.1.

TODO Continue analysis.

3.2 Pin Control Protection

TODO Pin Control Protection Design.

Pin Control Attack				
Precondition	*Vulnerability*	*Target*	*Attack method*	*Effect*
Local or Remote Access, Miscellaneous	Unknown	Device	Malware	Unknown

Table 3.1. Pin Control Attack classification based on Papp et al. [4] taxonomy.

Chapter 4

Experimental Results

TODO Experimental Results.

Chapter 5

Conclusions

TODO Conclusions.

Bibliography

[1] A.Abbasi, M.Hashemi, E.Zambon, S.Etalle, "Stealth Low-Level Manipulation of Programmable Logic Controllers I/O By Pin Control Exploitation", TODO CRITIS Conference 2016.

[2] N.Falliere, L.O Murchu, E.Chien, "W32.Stuxnet Dossier", Version 1.4, Febr. 2011. Online: `http://www.symantec.com/content/en/us/enterprise/media/security_response/whitepapers/w32_stuxnet_dossier.pdf`.

[3] R.Grandgenett, W.Mahoney, R.Gandhi, "Authentication Bypass and Remote Escalated I/O Command Attacks", CISR '15: Proceedings of the 10th Annual Cyber and Information Security Research Conference, Oak Ridge, Tennesse (USA), April 7-9, 2015, DOI 10.1145/2746266.2746268.

[4] D.Papp, Z.Ma, L.Buttyan, "Embedded systems security: Threats, vulnerabilities, and attack taxonomy" Privacy, Security and Trust (PST) 13th Annual Conference, Izmir (Turkey), July 21-23, 2015 pp. 145-152, DOI 10.1109/PST.2015.7232966.

[5] Z.Basnight, J.Butts, J.Lopez Jr., T.Dube, "Firmware modification attacks on programmable logic controllers", International Journal of Critical Infrastructure Protection, Volume 6, Issue 2, June 2013, pp. 76-84, DOI 10.1016/j.ijcip.2013.04.004.

[6] D.Peck, D.Peterson, "Leveraging ethernet card vulnerabilities in field devices", Proceedings of the SCADA Security Scientific Symposium, Miami Beach, Florida (USA), Jan. 18-19, 2009, pp. 1-19.

[7] A.Cui, M.Costello, S.J.Stolfo, "When Firmware Modifications Attack: A Case Study of Embedded Exploitation", 20th Annual Network & Distributed System Security Symposium, San Diego, California (USA), Febr. 24-27, 2013, DOI 10.7916/D8P55NKB.

[8] S.McLaughlin, "On Dynamic Malware Payloads Aimed at Programmable Logic Controllers", In 6th USENIX Workshop on Hot Topics in Security, 2011.

[9] S.McLaughlin, P.McDaniel, "SABOT: Specification-based Payload Generation for Programmable Logic Controllers", CCS '12: Proceedings of the 2012 ACM conference on Computer and Communications Security, New York (USA), Oct. 16-18, 2012, pp. 439-449, DOI 10.1145/2382196.2382244.

[10] D.Beresford, "Exploiting Siemens Simatic S7 PLCs", Black Hat USA 2011, Las Vegas, Nevada (USA), Aug. 3-4, 2011.

[11] J.Klick, S.Lau, D.Marzin, J.Malchow, V.Roth, "Internet-facing PLCs as a Network Backdoor", Proceedings of IEEE Conference on Communications and Network Security (CNS), Florence (Italy), Sept. 28-30, 2015, pp. 524-532, DOI 10.1109/CNS.2015.7346865.

[12] ICS-CERT, "Schneider Electric Modicon M340 Buffer Overflow Vulnerability", Dec. 17, 2015. Online: `https://ics-cert.us-cert.gov/advisories/ICSA-15-351-01`.

[13] ICS-CERT, "Rockwell Automation Micrologix 1100 and 1400 PLC Systems Vulnerabilities (Update A)", Oct. 27, 2015. Online: `https://ics-cert.us-cert.gov/advisories/ICSA-15-300-03A`.

[14] ICS-CERT, "Rockwell Automation MicroLogix 1100 PLC Overflow Vulnerability", Jan. 26, 2016. Online: `https://ics-cert.us-cert.gov/advisories/ICSA-16-026-02`.

[15] ICS-CERT, "Eaton ELCSoft Programming Software Memory Vulnerabilities", June 30, 2016. Online: `https://ics-cert.us-cert.gov/advisories/ICSA-16-182-01`.

[16] P.Chen, X.Xing, B.Mao, L.Xie, "Return-oriented rootkit without returns (on the x86)" ICICS 2010: 12th International Conference on Information and Communications Security, Barcelona (Spain), Dec. 15-17, 2010, pp. 340-354, DOI 10.1007/978-3-642-17650-0_24.

[17] S.Checkoway, L.Davi, A.Dmitrienko, A.Sadeghi, H.Shacham, M.Winandy, “Return-Oriented Programming without Returns”, CCS ’10: Proceedings of the 17th ACM conference on Computer and Communications Security, Chicago, Illinois (USA), Oct. 4-8, 2010, DOI 10.1145/1866307.1866370.